Categorical Phase Semantics for Linear Logic

Riley Shahar[1] and Steve Zdancewic[2]

[1] Reed College,
rileyshahar@reed.edu
[2] University of Pennsylvania,
stevez@cis.upenn.edu

Abstract. We introduce the generalized phase semantics of linear logic, a categorical framework for constructing provability-sound models of linear logic from the subobject posets of monoids. We prove soundness and completeness results in categories with well-behaved factorization systems, and prove soundness for fragments of the logic in categories with less structure present.

Keywords: Linear Logic, Categorical Models, Phase Semantics

1 Introduction

Linear logic, due to Girard [1], is a logic suitable for situations in which resources are constrained. It generalizes classical and intuitionistic logic by abandoning the structural rules of weakening and contraction, hence ensuring that hypotheses are used once and only once. The logic has widespread applications to resource-constrained settings, including to parallel computation [2], memory management [3], linguistics [4], and quantum mechanics [5]. Denotational semantics of linear logic have also been well-studied, including in coherence spaces [1], "dialogue" games [6], proof nets [7], finite vector spaces [8], quantales [9], and Chu spaces [10].

An especially important model of linear logic is Girard's phase semantics [1], which produce such a model from any (set-theoretic) monoid. Among many applications, they have been used to prove technical lemmas about linear logic, such as completeness [1] and cut elimination [11]; to show undecidability of higher-order linear logic [12] and boolean bunched implications [13]; and to establish safety properties of programs in the linear concurrent constraint calculus [14]. A central virtue in these applications is the *tunability* of the semantics: monoids are such ubiquitous objects that a phase semantics can often be found in close connection to any resource-constrained problem.

In the literature, it is common to construct ad-hoc varieties of phase semantics to fit a given use-case [1,11,13,14,15,16,17]. These semantics often enrich the set-theoretic monoid with extra axioms or structure: for instance, Larchey-Wendling and Galmiche allow the monoid multiplication to be non-deterministic [13], and Hamano and Takemura affix an interior operator to the monoid [16].

Meanwhile, monoid objects, a categorical generalization of monoids, are even more ubiquitous in computer science [18]. Well-known computational examples include monads [19], arrows [20], and graded monads [21]. As a result, a natural aim is to use monoid objects to bridge monads and linear logic. Indeed, the relationship between

Extra Structure	Obtained Connective(s)
none	$\top, \otimes$
η a subobject	1
pullbacks	&
initial object	0
coproducts	$\oplus$
tensor preserves coproducts	$\multimap$, $\parr$, $\bot$

Table 1: Adding structure to monoidal categories with factorization.

monads and linear logic has been previously studied. For instance, Chen and Hudak give a CPS-style translation converting linear datatypes into instances of the state monad [22]. In this paper, we choose a more semantic route, generalizing phase semantics to monoid objects over categories with certain structure.

Specifically, this paper makes the following contributions:

- We show that the subobject posets of monoids in categories with well-behaved factorization systems form models of linear logic. The construction very closely follows Girard's original construction of phase semantics, abstracting away from elements in favor of a product native to the subobjects.
- By asking for additional structure from the underlying category, we obtain different fragments of the logic, as detailed in Table 1. For instance, when the category has coproducts, we obtain the *positive* $(\otimes, \oplus)$ fragment.

One of our central technical constructions produces a quantale from the subobjects posets of monoids. This fact was first noticed by Niefield in an algebraic setting [23], though they do not explicitly construct the product, and do not relate their work to logic. In this light, our work can be seen as relating Niefield's construction to Yetter's quantale semantics of linear logic [9]. This relationship provides an easy way to instanstiate interesting models of linear logic, providing an avenue for both developing novel models and applying phase semantics to specific problem domains.

In Section 2, we introduce the background in category theory and linear logic needed for our technical work. Section 3 then defines the subobject product and join, a slight generalization of Niefield's construction necessary to specialize our semantics to different fragments of linear logic. In Section 4 we define the generalized phase semantics and give soundness and completeness theorems. In Section 5 we discuss related work and potential applications.

2 Preliminaries

2.1 Categorical Notions

We briefly recall some categorical context; a standard reference is Mac Lane & Moerdijk [24].

Functor Categories. Where $\mathcal{C}$ and $\mathcal{D}$ are categories, the *functor category* $\mathcal{D}^{\mathcal{C}}$ has functors $\mathcal{C} \to \mathcal{D}$ as objects and natural transformations between these functors as morphisms, with (vertical) composition of natural transformations as the composition. Limits in functor categories are computed pointwise, i.e. if the category $\mathcal{D}$ has limits of a certain shape, then they assemble functorially into the same limits in $\mathcal{D}^{\mathcal{C}}$.

Monoidal Categories. A *monoidal category* $\mathcal{K}$ supplements the categorical structure with a functorial *tensor product* $\otimes : \mathcal{K} \times \mathcal{K} \to \mathcal{K}$ and a *tensor unit* I, together with natural isomorphisms witnessing the associativity of $\otimes$ and unitality of I, and coherence laws that guarantee that composites of these isomorphisms are unique. For example, Set is monoidal with the cartesian product and the singleton, and any endofunctor category is monoidal with the (horizontal) composite and the identity functor.

A *monoid object* in a monoidal category is an object m together with morphisms $\mu : m \otimes m \to m$ and $\eta : I \to m$ such that μ associates and η is a unit of μ. For example, the monoid objects internal to Set are precisely classical monoids, and monoid objects internal to any endofunctor category $\mathcal{C}^{\mathcal{C}}$ are monads over $\mathcal{C}$.

A monoidal category is *braided* if there is a natural isomorphism witnessing commutativity of the tensor, satisfying certain coherence laws with the associator. It is *symmetric* if this braiding is involutive, i.e. if commuting twice in the same direction recovers the identity. A monoid object is *commutative* if its product is compatible with the braiding.

Subobjects. Fix a category $\mathcal{C}$ and an object x. Let $i : y \hookrightarrow x$ and $j : z \hookrightarrow x$ be monomorphisms. We say $i \leq j$ (or $y \leq z$, where the morphisms are clear) when i factors through j via a monomorphism ϕ. This relation gives a preorder on monomorphisms into x. If $i \leq j$ and $j \leq i$, then the morphisms ϕ are in fact isomorphisms, and we say $i \cong j$.

A *subobject* of x is an equivalence class of $\cong$. The class of subobjects of x forms a poset, the *subobject poset* $\mathrm{Sub}_{\mathcal{C}}(x)$ (we may omit the category when it's clear from context). By abuse of language, we often use the term subobject for a specific monomorphism $i : y \hookrightarrow x$, or, where the morphism is clear, even just a specific object y.

A *subfunctor* is just a subobject in a functor category. We can think of a subfunctor of the functor F as a (functorial) specification of a subobject of Fx for each object x in the domain. For example, a subobject of a set X is a class of injections $i : Y \hookrightarrow X$ whose images agree. In this way, subsets of X are in bijection with $\mathrm{Sub}_{\mathrm{Set}}(X)$. Further, working over the endofunctor category $\mathrm{Set}^{\mathcal{C}}$, up to equivalence we can choose a canonical representative G of any subfunctor of F such that $Gx \subseteq Fx$ for each x.

Factorization. Over Set, any function can be factored, uniquely up to unique isomorphism, into an epimorphism (surjection) followed by a monomorphism (injection). If this property holds in a category $\mathcal{C}$, we say $\mathcal{C}$ has *epi-mono factorization*. We call the monomorphism the *image* of the original morphism.

2.2 Linear Logic

A Linear Sequent Calculus. Here we give a standard sequent calculus for linear logic, following Troelstra [25], mostly to fix a notational convention.

The syntax of linear logic formulas is built inductively from uninterpreted base types; the nullary connectives (units) $0, 1, \bot, \top$; the unary connectives $!, ?, (-)^{\bot}$; and the binary connectives $\otimes, \oplus, \&, \parr, \multimap$. We give a two-sided presentation of the linear sequent calculus in Table 2. The linear dual $(-)^\perp$ of a formula is inductively defined in the standard way; linear implication may equivalently be defined as $A \multimap B := A^\perp \parr B$.

The intuitionistic fragment of linear logic consists of linear logic without $\parr$, $\perp$, $(-)^\perp$, ?, and the rule 0R. In the two-sided sequent presentation, this has a straightforward interpretation as only allowing one term on the right side of the sequent. Other fragments are defined in the standard way, by taking certain clases of connectives as suggested by the captions in Table 2. For example, the intuitionistic multiplicative-additive fragment consists of some base types together with $\otimes, \&, \oplus, \top, 1, 0$, and $\multimap$.

$$\frac{}{A \vdash A}\,\textsc{id} \qquad \frac{\Gamma, A, B, \Gamma' \vdash \Delta}{\Gamma, B, A, \Gamma' \vdash \Delta}\,\textsc{exc} \qquad \frac{\Gamma, A_i \vdash \Delta}{\Gamma, A_1 \,\&\, A_2 \vdash \Delta}\,\&_i\mathrm{L} \qquad \frac{\Gamma \vdash A, \Delta \qquad \Gamma \vdash B, \Delta}{\Gamma \vdash A \,\&\, B, \Delta}\,\&\mathrm{R}$$

$$\frac{\Gamma \vdash A \qquad A, \Delta \vdash B}{\Gamma, \Delta \vdash B}\,\textsc{cut} \qquad \frac{\Gamma, A \vdash \Delta \qquad \Gamma, B \vdash \Delta}{\Gamma, A \oplus B \vdash \Delta}\,\oplus\mathrm{L} \qquad \frac{\Gamma \vdash A_i, \Delta}{\Gamma \vdash A_1 \oplus A_2, \Delta}\,\oplus_i\mathrm{R}$$

Structural rules Additives

$$\frac{\Gamma, A, B \vdash \Delta}{\Gamma, A \otimes B \vdash \Delta}\,\otimes\mathrm{L} \qquad \frac{\Gamma \vdash A, \Delta \qquad \Gamma' \vdash B, \Delta'}{\Gamma, \Gamma' \vdash A \otimes B, \Delta, \Delta'}\,\otimes\mathrm{R}$$

$$\frac{\Gamma \vdash \Delta}{\Gamma, 1 \vdash \Delta}\,1\mathrm{L} \qquad \frac{}{\vdash 1}\,1\mathrm{R}$$

$$\frac{}{\Gamma, \perp \vdash \Delta}\,\perp \qquad \frac{}{\Gamma \vdash \top, \Delta}\,\top \qquad \frac{\Gamma, A \vdash \Delta \qquad \Gamma', B \vdash \Delta'}{\Gamma, \Gamma', A \parr B \vdash \Delta, \Delta'}\,\parr\mathrm{L} \qquad \frac{\Gamma \vdash A, B, \Delta}{\Gamma \vdash A \parr B, \Delta}\,\parr\mathrm{R}$$

$$\frac{}{0 \vdash}\,0\mathrm{L} \qquad \frac{\Gamma \vdash \Delta}{\Gamma \vdash 0, \Delta}\,0\mathrm{R} \qquad \frac{\Gamma \vdash A, \Delta \qquad \Gamma', B \vdash \Delta'}{\Gamma, \Gamma', A \multimap B \vdash \Delta, \Delta'}\,\multimap\mathrm{L} \qquad \frac{\Gamma, A \vdash B, \Delta}{\Gamma \vdash A \multimap B, \Delta}\,\multimap\mathrm{R}$$

Units Multiplicatives

Table 2: A linear sequent calculus
(A and B range over formulas; Γ and Δ range over sequents).

Phase Semantics. Phase semantics are Girard's original provability-complete semantics of linear logic [1]. A *phase space* $(M, e, \perp)$ consists of a a commutative classical monoid M with unit e and a subset $\perp \subseteq M$. We define operations on subsets of M as in Table 3. Propositions are interpreted as *facts*, meaning subsets in the image of $(-)^\perp$.

Quantale Semantics. Quantale semantics, due to Yetter [9], are an "element-free" phase semantics. We first need some basic notions from quantale theory [26].

Definition 1. *A* quantale Q *is a complete lattice with a unital, associative binary operation* $*$ *which is compatible with the order and distributes over arbitrary joins.*

Example 1. For any monoid M, the powerset $\mathcal{P}(M)$ is a quantale, with join and meet given by union and intersection, and $*$ given by the direct product of two subsets, $AB =$

$$X^\perp := \{y \in M : \forall x \in X, xy \in \perp\} \qquad [\![A \otimes B]\!] := \{xy : x \in [\![A]\!], y \in [\![B]\!]\}^{\perp\perp}$$

$$[\![A \parr B]\!] := ([\![A]\!]^\perp \otimes [\![B]\!]^\perp)^\perp \qquad [\![A \oplus B]\!] := ([\![A]\!] \cup [\![B]\!])^{\perp\perp} \qquad [\![A \,\&\, B]\!] := [\![A]\!] \cap [\![B]\!]$$

$$[\![0]\!] := \varnothing^{\perp\perp} \qquad [\![\top]\!] := M \qquad [\![1]\!] := \{e\}^{\perp\perp} \qquad [\![!A]\!] := ([\![A]\!] \cap I)^{\perp\perp} \qquad [\![?A]\!] := ([\![A]\!]^\perp \cap I)^\perp.$$

Table 3: The phase semantics of linear logic
(I is the set of idempotents in $[\![1]\!]$).

$\{ab : a \in A, b \in B\}$. In fact, the key observation of quantale semantics is that none of the operations in Table 3 require elements to define.

Definition 2. *A* quantic nucleus $(-)^\diamond$ *is a lax monoidal monad on a quantale. Explicitly, it is monotone, idempotent, inflationary (i.e. a closure operator), and satisfies*

$$x^\diamond * y^\diamond \leq (x * y)^\diamond.$$

Example 2. The double negation $(-)^{\perp\perp}$ of phase semantics is a nucleus on the powerset.

Proposition 1. *In a quantale, since the product preserves joins, the functors $(- * b)$ and $(b * -)$ are left adjoints.*

For simplicity, we restrict ourselves commutative quantales, in which case the right adjoints agree. The more general case is discussed by Yetter.

Definition 3. *We write*

$$x \multimap y := \bigvee \{z : x * z \leq y\}.$$

for the right adjoint to $*$. In quantale theory, $\multimap$ is called a residuation*.*

Definition 4. *A element $\perp$ of a quantale is* dualizing *if $(x \multimap \perp) \multimap \perp = x$ for any x. A quantale with a dualizing element is a* Girard quantale*.*

Proposition 2. *The following hold:*

1. *The image of a quantale Q under a nucleus is a quantale with:*
   - *The same meet as Q.*
   - *Join and product given by the image of Q's join and product under the nucleus.*
2. *For any element x, $(- \multimap x) \multimap x$ is a nucleus.*
3. *The image of the above nucleus is a Girard quantale with dualizing element x.*

Yetter gives a semantics of linear logic in any quantale Q with specified element $\perp$ (which need not be dualizing, but will become dualizing when lifted to the induced Girard quantale). The key observation is that Girard quantales naturally model linear logic, and any quantale induces a Girard quantale by Proposition 2. We explicitly define the operations on the original quantale in Table 4; the phase semantics of a monoid M are exactly the quantale semantics of the quantale $\mathcal{P}(M)$.

$$(x)^\perp := x \multimap \perp \qquad [\![A \otimes B]\!] := ([\![A]\!] * [\![B]\!])^{\perp\perp} \qquad [\![A \parr B]\!] := ([\![A]\!]^\perp * [\![B]\!]^\perp)^\perp$$

$$[\![A]\!] \oplus [\![B]\!] := ([\![A]\!] \vee [\![B]\!])^{\perp\perp} \qquad [\![A \,\&\, B]\!] := [\![A]\!] \wedge [\![B]\!] \qquad [\![0]\!] := (\bigwedge Q)^{\perp\perp}$$

$$[\![\top]\!] := \bigvee Q \qquad [\![1]\!] := e^{\perp\perp}$$

Table 4: The quantale semantics of linear logic.

3 The Subobject Product

In this section, we give our first technical contribution, which enables constructing a quantale structure from the subobjects of monoid objects in certain categories; when less categorical structure is present, we obtain only some of this algebraic structure. Fix a monoidal category $(C, \otimes, I)$ with factorization and a monoid object (m, μ, η) in C.

3.1 Monoidal Structure

Let $i : x \hookrightarrow m$ and $j : y \hookrightarrow m$ be subobjects of m. There is a canonical morphism

$$x \otimes y \xrightarrow{i \otimes j} m \otimes m \xrightarrow{\mu} m. \tag{1}$$

Definition 5 (Subobject Product). *The* subobject product $i * j : x * y \hookrightarrow m$ *is the image of the morphism* (1).

Example 3. Let the ambient category be Set, so that a monoid object M is just a classical monoid, and subobjects are just subsets. The tensor takes the catesian product, so the morphism (1) takes the pairwise product of each element of the subsets. After taking the image we recover the familiar direct product, $X * Y \cong XY = \{xy : x \in X, y \in Y\}$.

Example 4. Consider the (covariant) powerset monad $\mathcal{P}$, whose multiplication is given by the union. For any natural number n, there is a subfunctor $\mathcal{P}_n$ which sends a set to the set of its subsets with at most n elements. The tensor is just functor composiiton, so $\mathcal{P}_n \otimes \mathcal{P}_m$ takes a set X to the set of sets of at most n sets of at most m elements of X. Taking the union yields sets of at most nm elements, but not injectively. For example, consider (1) instantiated as $\mathcal{P}_2\mathcal{P}_2\{1,2,3\} \to \mathcal{P}\{1,2,3\}$; the set $\{1,2\}$ is the image of both $\{\{1,2\}\}$ and $\{\{1\},\{2\}\}$. However, taking the image recovers a functor isomorphic to $\mathcal{P}_{nm}$.

Proposition 3. *If $\otimes$ preserves epimorphisms, then the subobject product is associative.*

We need the hypothesis of Proposition 3 to ensure that the morphisms $(x \otimes y) \otimes z \to (x * y) \otimes z$ and $x \otimes (y \otimes z) \to x \otimes (y * z)$ are epis. Fortunately, it's relatively minor: it's satisfied, for example, if the monoidal structure is symmetric closed, since then $\otimes$ is a left adjoint in both arguments and so preserves colimits.

Proposition 4. *If the monoid unit η is a subobject, then the subobject product is unital with identity η.*

Proposition 5. *If the ambient category is symmetric monoidal and m is commutative, then the subobject product is commutative.*

Proposition 6. *The subobject product preserves subobject inclusion.*

Propositions 3 and 4 turn $\mathrm{Sub}_C(m)$ into a monoidal category; Proposition 5 makes this monoidal category symmetric. Proposition 6 makes this into an ordered monoid.

3.2 Quantale Structure

Now, let the underlying category have coproducts, and fix two subobjects $i : x \hookrightarrow m$ and $j : y \hookrightarrow m$. Again, by the universal property of the coproduct, we have a canonical morphism

$$x + y \xrightarrow{i+j} m. \tag{2}$$

Definition 6 (Subobject Join). *The* subobject join $i \vee j : x \vee y \hookrightarrow m$ *is the image of the morphism* (2).

Multi-place and infinitary joins are defined similarly; associativity follows from that of coproducts. Note that Definition 6 makes sense even when m is not a monoid object.

Example 5. Again, let the ambient category be Set. The coproduct of two sets X and Y is their disjoint union. When X and Y are subsets of M, the map (2) is injective exactly if X and Y are disjoint. If not, the quotient equates elements that include to the same element of M, i.e. elements of the disjoint union that represent the same underlying element. The join is therefore just the union of subsets, as expected.

The following two results are due to Niefield [23]:

Proposition 7. *The subobject join is a join with respect to subobject inclusion.*

Proposition 8. *If $\otimes$ preserves coproducts, then the subobject product distributes over arbitrary subobject joins.*

Finally, a standard result is that:

Proposition 9. *When C has pullbacks, the pullback $x \wedge y$ of two subobjects is their meet with respect to subobject inclusion.*

In other words, the subobject poset is a quantale. In particular, $*$ has a right adjoint, $\multimap$.

4 Generalized Phase Semantics

Following Yetter [9], a semantics of linear logic may be instantiated from any quantale. In this section, we adapt Yetter's construction to the structure induced by the subobject product and join over monoid subobjects.

As in Definition 3, for notational simplicity we will restrict ourselves to the case of symmetric monoidal categories and commutative monoids, in which case we write $\multimap$ for the right adjoint to the quantale multiplication. Our results extend to the non-commutative case easily but with some notational overhead; in this case we obtain a variant of non-commutative linear logic.

4.1 Definition

Yetter limits the semantics to quantales, but the definition of a quantic nucleus, phrased categorically as in Definition 2, makes sense in the much more general setting of monoidal posets. That observation motivates the following definition:

Definition 7. *Let $\mathcal{C}$ be a symmetric monoidal category with factorization, and let (m, μ, η) be a commutative monoid in $\mathcal{C}$. Let $(-)^{\diamond}$ be a lax monoidal monad on the monoidal poset* $\mathrm{Sub}_{\mathcal{C}}(m)$*. We call $(m, \diamond)$ an* intuitionistic generalized phase space*.*

A subobject is a fact *if it is in the image of $\diamond$. Given some base types, an instance of* intuitionistic generalized phase semantics *over $(m, \diamond)$ consists of, for each base type x, a fact $[\![x]\!]$. Define:*

$$[\![A \otimes B]\!] := ([\![A]\!] * [\![B]\!])^{\diamond}, \qquad [\![\top]\!] := m.$$

If η is a subobject of m, define:

$$[\![1]\!] := \eta^{\diamond}.$$

If $\mathcal{C}$ has pullbacks, define:

$$[\![A \,\&\, B]\!] := [\![A]\!] \wedge [\![B]\!].$$

If $\mathcal{C}$ has an initial object $\varnothing$ (which is then always a subobject), define:

$$[\![0]\!] := \varnothing^{\diamond}.$$

If $\mathcal{C}$ has coproducts, define:

$$[\![A \oplus B]\!] := ([\![A]\!] \vee [\![B]\!])^{\diamond}.$$

If $\mathcal{C}$ has coproducts which are preserved by $\otimes$, define:

$$[\![A \multimap B]\!] := [\![A]\!] \multimap\!\!\bullet\; [\![B]\!].$$

Throughout the paper, but especially in Definition 7, we have been imprecise about distinguishing between inclusions, subobjects, and the domains of inclusions. Formally, the subobject is a certain equivalence class of monomorphisms into m, and $\diamond$ is a functor on the subobject poset (not on the base category), so it preserves equivalence classes by definition. As such, all these notions are well-defined.

We can easily extend Definition 7 to the classical case, as follows.

Definition 8. *Let $\mathcal{C}$ be a symmetric monoidal category with coproducts and factorization such that $\otimes$ preserves coproducts, and let (m, μ, η) be a commutative monoid in $\mathcal{C}$. Let $\bot$ be a subobject of m. We call $(m, \bot)$ a* generalized phase space*. In this setting, m is a quantale, and the* generalized phase semantics *over $(m, \bot)$ are just the quantale semantics of m.*

Explicitly, this quantale semantics are instantiated as follows. Let

$$[\![A^{\bot}]\!] := [\![A]\!] \multimap\!\!\bullet\; \bot.$$

A fact *is any subobject in the image of $(-)^{\bot}$. Define the remaining intuitionistic connectives as in Definition 7 with $\diamond = (-)^{\bot\bot}$. Finally, define:*

$$[\![\bot]\!] = \bot, \qquad [\![A \parr B]\!] := ([\![A]\!]^{\bot} * [\![B]\!]^{\bot})^{\bot}.$$

4.2 Completeness Theorems

The classical phase semantics of linear logic are known to be complete. In this section, we show that these are a special case when our semantics are instantiated in the category of sets, and hence we obtain completeness for generalized phase semantics.

Theorem 1. *Generalized phase semantics in* Set *are exactly phase semantics.*

Proof. Most of the work is already done. In Example 3, we showed that the subobject product is the direct product of subsets. In Example 5, we showed that the subobject join is the union of subsets. We have that the meet of subsets is the intersection. It remains to show that the residuation $X \multimap \perp$ yields the same linear negation as the formula for $X^\perp$ from Table 3. We have

$$X^\perp = \{y \in M : \forall x \in X, xy \in \perp\} = \bigcup \{Y \subseteq M : XY \subseteq \perp\},$$

which is exactly the formula for the residuation in Definition 3.

In fact, this theorem is sufficient to obtain quantale semantics as a special case:

Corollary 1. *Every instance of quantale semantics is isomorphic to an instance of generalized phase semantics.*

Proof. This follows from a result of Rosenthal [27] that every Girard quantale is isomorphic to an instance of phase semantics.

Similarly, we straightforwardly obtain Abrusci's intuitionistic phase semantics [15] by instantiating intuitionistic generalized phase semantics in Set.

With this result, we immediately obtain completeness.

Theorem 2 (Completeness). *Generalized phase semantics are complete for provability in linear logic. Explicitly, if* $[\![A_1 \otimes \cdots \otimes A_m]\!] \leq [\![B_1 \parr \cdots \parr B_n]\!]$ *in all generalized phase spaces, then* $A_1, \ldots, A_m \vdash B_1, \ldots, B_n$ *is provable.*

Proof. Since phase semantics are known to be complete—in particular, they can be instantiated with the free monoid on the syntax—we inherit completeness.

4.3 Soundness Theorems

We now turn to soundness. As with completeness, in Theorem 3 we obtain a straightforward soundness theorem. As before, with more care we can be more precise. In particular, we show in Theorem 4 that in any category where certain operations can be constructed, the semantics are sound for the corresponding fragment of intuitionistic linear logic. For example, we show that generalized phase semantics in SMCs with coproducts are sound for the $(\otimes, \oplus)$ fragment.

Theorem 3 (Soundness). *Generalized phase semantics are sound for provability in linear logic. Explicitly, if* $A_1, \ldots, A_m \vdash B_1 \ldots, B_n$ *is provable, then* $[\![A_1 \otimes \cdots \otimes A_m]\!] \leq [\![B_1 \parr \cdots \parr B_n]\!]$ *in all generalized phase spaces.*

Proof. Generalized phase semantics are a special case of quantale semantics, which are known to obtain the same soundness result.

For the more specialized result, we first notice that the denotation of any formula is a fact. This observation allows lifting useful results from the subobject poset to the poset of facts.

Lemma 1. *Let A be a formula of intuitionistic linear logic. Then in any intuitionistic generalized phase space where $[\![A]\!]$ can be constructed, $[\![A]\!]$ is a fact, i.e. $[\![A]\!] = a^\diamond$ for some subobject a.*

Proof. The proof is by induction on the structure of A. Most of the cases follow because $\diamond$ is in the definition of the denotation. The other cases are as follows:

Atoms. By construction, base types are always denoted as facts.
$\top$. Since $\diamond$ is inflationary, $m = m^\diamond$.
&. Let $A = a^\diamond$ and $B = b^\diamond$. Then since $\wedge$ is a meet and $\diamond$ preserves the order,

$$(A \wedge B)^\diamond = (a^\diamond \wedge b^\diamond)^\diamond \leq a^{\diamond\diamond} = A^\diamond = A,$$

and similarly for B. Thus $(A \wedge B)^\diamond \leq A \wedge B$ by definition of the meet, and so since $\diamond$ is inflationary $(A \wedge B)^\diamond = A \wedge B$, as desired.
$\multimap$. In this case we have a full quantale; this is a standard result in quantale theory.

Theorem 4 (Fragmentary Soundness). *Any instance of intuitionistic generalized phase semantics is sound for provability in the appropriate fragment of intuitionistic linear logic. Explicitly, if $A_1, \ldots, A_n \vdash B$ is provable, then $[\![A_1 \otimes \cdots \otimes A_n]\!] \leq [\![B]\!]$ in all intuitionistic generalized phase spaces where the relevant connectives can be constructed, as outlined in Table 1.*

Proof. We proceed by induction on the structure of a derivation in the sequent calculus of Table 2. Recall that in the intuitionistic case right-sided sequents are singletons.

Id. Every subobject includes into itself by the identity.
Exc. This is the commutativity of $*$ from Proposition 5.
Cut. We have $\Gamma \leq A$ and $A * \Delta \leq B$[3]. Since $*$ preserves the order,

$$[\![\Gamma \otimes \Delta]\!] = \Gamma * \Delta \leq A * \Delta \leq B = [\![B]\!].$$

$\otimes$L. Since left-sided sequents are interpreted as tensors, this is reflexivity.
$\otimes$R. The assumptions are $\Gamma \leq A$ and $\Gamma' \leq B$. Since both $*$ and $\diamond$ preserve the order,

$$[\![\Gamma \otimes \Gamma']\!] = (\Gamma * \Gamma')^\diamond \leq (A * \Gamma')^\diamond \leq (A * B)^\diamond = [\![A \otimes B]\!].$$

[3] Here, and throughout the remainder of the proof, we will be loose with the distinction between a sequent Γ and its denotation $[\![\Gamma]\!]$, a subobject. For example, the display under **$\otimes$R** should read $[\![\Gamma \otimes \Gamma']\!] = ([\![\Gamma]\!] * [\![\Gamma']\!])^\diamond \leq ([\![A]\!] * [\![\Gamma']\!])^\diamond \leq ([\![A]\!] * [\![B]\!])^\diamond = [\![A \otimes B]\!]$. One way to see this is as overloadingly lifting the operators on the subobjects to operators on the syntax.

&ᵢL. The assumption is that $\Gamma \otimes A_i \leq \Delta$. Since $\wedge$ is a meet and $*$ preserves the order,

$$[\![\Gamma \otimes (A_1 \,\&\, A_2)]\!] = \Gamma * (A_1 \wedge A_2) \leq \Gamma * A_i \leq \Delta = [\![\Delta]\!].$$

&R. Since & is the meet of facts, the result follows.
⊕L. Since $\oplus$ is the join of facts, the result follows.
$\oplus_i$R. The assumption is that $\Gamma \leq A_i$. Since $\vee$ is a join and $\diamond$ preserves the order,

$$[\![\Gamma]\!] \leq A_i \leq A_1 \vee A_2 \leq (A_1 \vee A_2)^\diamond = [\![A_1 \oplus A_2]\!].$$

⊸L. The assumptions are that $\Gamma \leq A$ and $\Gamma' * B \leq \Delta$. We have

$$(\Gamma * \Gamma' * (A \multimap\!\bullet\, B))^\diamond \leq (A * \Gamma' * (A \multimap\!\bullet\, B))^\diamond \leq (\Gamma' * B)^\diamond \leq \Delta^\diamond = \Delta,$$

where $A * (A \multimap\!\bullet\, B) \leq B$ is the unit of the adjunction.
⊸R. This is one direction of the definition of an adjunction.
1L. The assumption is that $\Gamma \leq \Delta$. Let $\Gamma = \gamma^\diamond$. Now

$$[\![\Gamma \otimes 1]\!] = (\gamma^\diamond * \eta^\diamond)^\diamond = (\gamma * \eta)^\diamond = \gamma^\diamond = \Gamma \leq \Delta = [\![\Delta]\!].$$

1R. The empty left side is interpreted as the unit of $\otimes$, i.e. 1, so this is reflexivity.
⊤. Since Δ is empty in the intuitionistic case, this is immediate from the definition.
0L. This follows from minimality of $\varnothing$, since $\diamond$ preserves the order.

To see the result over categories with only some of the necessary structure, it suffices to notice that the induction above only relies on the structure necessary to construct each connective to obtain the corresponding branch of the proof.

5 Discussion

As mentioned, our construction in Section 3 is closely related to a result of Niefield [23] that monoid subobjects form a quantale in suitable categories. We have discussed this result at the end of the introduction, but it is worth reiterating that our construction agrees with Niefield's when all the necessary structure is present, but still goes through when only some of the structure is present.

There are several algebraic frameworks for provability-complete semantics of linear logic known in the literature. IL-algebras [25] are in some sense a more natural semantics for linear logic than quantales, since morphisms of IL-algebras more directly preserve the semantic structure. IL-algebras are often used to prove soundness of other semantics, for example for non-deterministic phase semantics [13]. However, unlike generalized phase semantics, these algebraic frameworks do not prescribe constructions of the semantics, but instead just provide an axiomatization of the necessary structure. Furthermore, we are not aware of an algebraic framework which obtains the modularity of Theorem 4: these frameworks often have separete intuitionistic, classical, and exponential axiomatizations, but do not have soundness proofs for each connective in isolation. Generalized phase semantics should be seen as a toolkit for actively constructing new semantics, rather than a framework for proving soundness of already-instantiated models.

When generalized phase semantics are instantiated in endofunctor categories, as in Example 4, the subobject product gives a kind of composition of subfunctors of monads. Similarly, Eklund *et al.* give a strategy for composing submonads of monads [28]. However, while our construction allows unconditionally composing subfunctors of the same monad, theirs gives conditions under which two submonads of different monads compose.

The phase semantics of other substructural logics, such as affine and relevance logic, have been studied [29]. Future work on generalized phase semantics should include extending the construction to these logics. Important questions include under what conditions the completeness and soundness results extend, and whether contraction or weakening obtain in certain categories or monoids even with our current construction.

Meanwhile, there are several interesting potential applications of our construction. For instance, two standard frameworks for analyzing probabalistic computation are the Giry monad [30] and linear logic [31]. Since monads are monoid objects, studying the generalized phase semantics of the Giry monad may provide a bridge between these distinct settings. However, there are potential issues with this approach; subobjects in functor categories are functors, which is often too much structure for the obtained semantics to be interesting, since most functors of interest contain the identity.

We have not been as strict as possible with the required structure for our semantics. For instance, it is common to consider factorization systems where one or both classes are more specific than mono/epimorphisms, such as regular or strong mono/epimorphisms. These systems give rise to well-behaved notions subobject posets, and are often more natural when working outside of a topos. Future work should generalize our results to these settings.

Finally, as future work we would like to obtain a more general completeness theorem, which gives conditions under which the generalized phase semantics of a category $\mathcal{C}$ are complete. Such a proof would likely proceed by embedding $\mathcal{C}$ in Set and reflecting enough of the structure of the free syntactic monoid back onto $\mathcal{C}$. This result, together with our soundness proof, would further simplify the construction of new semantics.

References

1. Girard, J.Y.: Linear logic. Theoretical Computer Science **50**(1) (1987) 1–101
2. Bucciarelli, A., Ehrhard, T., Manzonetto, G.: A relational semantics for parallelism and non-determinism in a functional setting. Annals of Pure and Applied Logic **163**(7) (2012) 918–934 The Symposium on Logical Foundations of Computer Science 2009.
3. Chirimar, J., Gunter, C.A., Riecke, J.G.: Proving memory management invariants for a language based on linear logic. In: Proceedings of the 1992 ACM Conference on LISP and Functional Programming. LFP '92, New York, NY, USA, Association for Computing Machinery (1992) 139–150
4. Moot, R., Piazza, M.: Linguistic applications of first order intuitionistic linear logic. Journal of Logic, Language and Information **10**(2) (2001) 211–232
5. Pratt, V.R.: Linear logic for generalized quantum mechanics. In: Proceedings Workshop on Physics and Computation, Dallas, IEEE Computer Society. (1993)
6. Blass, A.: A game semantics for linear logic. Annals of Pure and Applied Logic **56**(1) (1992) 183–220

7. Ehrhard, T.: An introduction to differential linear logic: proof-nets, models and antiderivatives. Mathematical Structures in Computer Science **28**(7) (2018) 995–1060
8. Valiron, B., Zdancewic, S.: Finite vector spaces as model of simply-typed lambda-calculi. In Ciobanu, G., Méry, D., eds.: Theoretical Aspects of Computing – ICTAC 2014, Cham, Springer International Publishing (2014) 442–459
9. Yetter, D.N.: Quantales and (noncommutative) linear logic. The Journal of Symbolic Logic **55**(1) (1990) 41–64
10. Pratt, V.: Chu spaces as a semantic bridge between linear logic and mathematics. Theoretical Computer Science **294**(3) (2003) 439–471 Linear Logic.
11. Okada, M.: Phase semantic cut-elimination and normalization proofs of first- and higher-order linear logic. Theoretical Computer Science **227**(1) (1999) 333–396
12. Lafont, Y.: The undecidability of second order linear logic without exponentials. The Journal of Symbolic Logic **61**(2) (1996) 541–548
13. Larchey-Wendling, D., Galmiche, D.: The undecidability of boolean bi through phase semantics. In: 2010 25th Annual IEEE Symposium on Logic in Computer Science. (2010) 140–149
14. Fages, F., Ruet, P., Soliman, S.: Linear concurrent constraint programming: Operational and phase semantics. Information and Computation **165**(1) (2001) 14–41
15. Abrusci, V.M.: Non-commutative intuitionistic linear logic. Mathematical Logic Quarterly **36**(4) (1990) 297–318
16. Hamano, M., Takemura, R.: A phase semantics for polarized linear logic and second order conservativity. Journal of Symbolic Logic **75**(1) (2010) 77 – 102
17. De, A., Jafarrahmani, F., Saurin, A.: Phase Semantics for Linear Logic with Least and Greatest Fixed Points. In: FSTTCS 2022 - 42nd IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science. Volume 250., Chennai, India (December 2022) 35:1–35:23
18. Rivas, E., Jaskelioff, M.: Notions of computation as monoids. Journal of Functional Programming **27** (2017) e21
19. Wadler, P.: Monads for functional programming. In Broy, M., ed.: Program Design Calculi, Berlin, Heidelberg, Springer Berlin Heidelberg (1993) 233–264
20. Jacobs, B., Heunen, C., Hasuo, I.: Categorical semantics for arrows. Journal of Functional Programming **19**(3-4) (2009) 403–438
21. Katsumata, S.y., McDermott, D., Uustalu, T., Wu, N.: Flexible presentations of graded monads. Proc. ACM Program. Lang. **6**(ICFP) (aug 2022)
22. Chen, C.P., Hudak, P.: Rolling your own mutable adt—a connection between linear types and monads. In: Proceedings of the 24th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages. POPL '97, New York, NY, USA, Association for Computing Machinery (1997) 54–66
23. Niefield, S.B.: Constructing quantales and their modules from monoidal categories. Cahiers de Topologie et Géométrie Différentielle Catégoriques **37**(2) (1996) 163–176
24. Mac Lane, S., Moerdijk, I.: Sheaves in Geometry and Logic. Springer New York (1994)
25. Troelstra, A.S.: Lectures on Linear Logic. Center for the Study of Language and Information Publications (1992)
26. Kruml, D., Paseka, J.: Algebraic and categorical aspects of quantales. Volume 5 of Handbook of Algebra. North-Holland (2008) 323–362
27. Rosenthal, K.I.: A note on girard quantales. Cahiers de Topologie et Géométrie Différentielle Catégoriques **31**(1) (1990) 3–11
28. Eklund, P., Galan, M., Medina, J., Ojeda-Aciego, M., Valverde, A.: Composing submonads. In: Proceedings 31st IEEE International Symposium on Multiple-Valued Logic. (2001) 367–372

29. Dal Lago, U., Martini, S.: Phase semantics and decidability of elementary affine logic. Theoretical Computer Science **318**(3) (2004) 409–433
30. Giry, M.: A categorical approach to probability theory. In Banaschewski, B., ed.: Categorical Aspects of Topology and Analysis, Berlin, Heidelberg, Springer Berlin Heidelberg (1982) 68–85
31. Ehrhard, T., Tasson, C.: Probabilistic call by push value. Logical Methods in Computer Science **15** (2019)

A Appendix: Proofs

Proof (of Proposition 3). Let $i : x \hookrightarrow m$, $j : y \hookrightarrow m$, and $k : z \hookrightarrow m$ be subobjects. The following diagram commutes (where α is the associator of $\otimes$, and applications of $\otimes$ have been elided for readability):

$$\begin{tikzcd}
& (xy)z \arrow[rr, "\alpha_{x,y,z}"] \arrow[dl, "(ij)1_z"'] \arrow[dd, "(ij)k"] & & x(yz) \arrow[dd, "i(jk)"'] \arrow[dr, "1_x(jk)"] & \\
(mm)z \arrow[d, "\mu 1_z"'] \arrow[dr, "1_{mm}k"] & & & & x(mm) \arrow[dl, "i1_{mm}"'] \arrow[d, "1_x\mu"] \\
mz \arrow[dr, "1_m k"'] & (mm)m \arrow[rr, "\alpha_{m,m,m}"] \arrow[d, "\mu 1_m"] & & m(mm) \arrow[d, "1_m\mu"'] & xm \arrow[dl, "i1_m"] \\
& mm \arrow[dr, "\mu"] & & mm \arrow[dl, "\mu"'] & \\
& & m. & &
\end{tikzcd}$$

Indeed, the top square commutes by naturality of α, all four side diagrams by bifunctoriality of $\otimes$, and the bottom pentagon by associativity of μ.

Now, taking images along the outside of this diagram, we obtain the following commutative diagram:

$$\begin{tikzcd}
& (xy)z \arrow[rr, "\alpha_{x,y,z}"] \arrow[dl, two heads] \arrow[d, "(ij)1_z"] & & x(yz) \arrow[d, "1_x(jk)"'] \arrow[dr, two heads] & \\
(x*y)z \arrow[dddd, two heads] \arrow[dr, "(i*j)1_z"'] & (mm)z \arrow[d, "\mu 1_z"] & & x(mm) \arrow[d, "1_x\mu"'] & x(y*z) \arrow[dl, "x(j*k)"] \arrow[dddd, two heads] \\
& mz \arrow[d, "1_m k"] & & xm \arrow[d, "i1_m"'] & \\
& mm \arrow[ddr, "\mu"] & & mm \arrow[ddl, "\mu"'] & \\
& & & & \\
(x*y)*z \arrow[rr, hook, "(i*j)*k"'] & & m & & x*(y*z). \arrow[ll, hook', "i*(j*k)"]
\end{tikzcd}$$

Since α is an isomorphism, the morphism $(xy)z \to m$ of this diagram factors through both $(x * y) * z$ and $x * (y * z)$; now we obtain the desired result by uniqueness of factorization.

Proof (of Proposition 4). Let $i : x \hookrightarrow m$ be a subobject. We have commutativity of the following diagram, by naturality of the left unitor λ and unitality of μ.

$$\begin{tikzcd}
Ix \arrow[r, "1_I i"] \arrow[dr, "\lambda"'] & Im \arrow[r, "\eta 1_m"] \arrow[dr, "\lambda"'] & mm \arrow[d, "\mu"] \\
& x \arrow[r, hook, "i"'] & m.
\end{tikzcd}$$

Since λ is an isomorphism and hence an epimorphism, x is the image of the top-right path along the diagram, giving the desired result by uniqueness of factorization. An identical argument works in the other direction.

Proof (of Proposition 5). Let $i : x \hookrightarrow m$ and $j : y \hookrightarrow m$ be subobjects. The following diagram commutes, by naturality of the braiding γ and commutativity of μ:

$$\begin{tikzcd}
xy \arrow[rr, "\gamma_{xy}"] \arrow[d, "ij"'] & & yx \arrow[d, "ji"] \\
mm \arrow[rr, "\gamma_{mm}"'] \arrow[dr, "\mu"'] & & mm \arrow[dl, "\mu"] \\
& m. &
\end{tikzcd}$$

Now $x * y$ and $y * x$ are both images of the morphism $xy \to m$ of this diagram, giving the desired result by uniqueness of factoriation.